Esercizio 1: Gestione di un negozio di elettronica Progettate un sistema per la gestione di un negozio di elettronica. Il negozio vende diverse categorie di prodotti, tra cui:

1. Product: un prodotto base con nome, descrizione, prezzo e quantità disponibile.
2. Electronics: un prodotto elettronico che eredita da Product e ha anche un produttore, un modello e una garanzia.
3. Computer: un computer che eredita da Electronics e ha anche una CPU, una quantità di RAM e una capacità di archiviazione.
4. Smartphone: uno smartphone che eredita da Electronics e ha anche un sistema operativo, una dimensione dello schermo e una fotocamera.
5. Accessory: un accessorio che eredita da Product e ha anche un tipo (ad esempio, custodia, caricabatterie, ecc.).

Ogni categoria di prodotto deve avere i seguenti metodi:

1. Product:
   * calculateTotalPrice(): calcola il prezzo totale del prodotto in base al prezzo unitario e alla quantità disponibile.
   * getProductInfo(): restituisce una stringa con le informazioni del prodotto (nome, descrizione, prezzo, quantità).
2. Electronics:
   * getWarrantyInfo(): restituisce una stringa con le informazioni sulla garanzia del prodotto elettronico.
   * getElectronicsInfo(): restituisce una stringa con le informazioni del prodotto elettronico (nome, descrizione, prezzo, quantità, produttore, modello, garanzia).
3. Computer:
   * getComputerSpecs(): restituisce una stringa con le specifiche del computer (CPU, RAM, archiviazione).
   * getComputerInfo(): restituisce una stringa con le informazioni del computer (nome, descrizione, prezzo, quantità, produttore, modello, garanzia, specifiche).
4. Smartphone:
   * getSmartphoneSpecs(): restituisce una stringa con le specifiche dello smartphone (sistema operativo, dimensione dello schermo, fotocamera).
   * getSmartphoneInfo(): restituisce una stringa con le informazioni dello smartphone (nome, descrizione, prezzo, quantità, produttore, modello, garanzia, specifiche).
5. Accessory:
   * getAccessoryType(): restituisce il tipo di accessorio.
   * getAccessoryInfo(): restituisce una stringa con le informazioni dell'accessorio (nome, descrizione, prezzo, quantità, tipo).

Inoltre, create una classe Store che gestisca un elenco di prodotti e fornisca i seguenti metodi:

* addProduct(Product): aggiunge un prodotto all'elenco.
* removeProduct(Product): rimuove un prodotto dall'elenco.
* searchByName(String): cerca un prodotto per nome e restituisce le informazioni del prodotto.
* calculateTotalInventoryValue(): calcola il valore totale dell'inventario del negozio.
* displayProductInfo(): stampa le informazioni di tutti i prodotti del negozio.

Infine, create un programma principale che istanzi diversi tipi di prodotti, li aggiunga all'elenco dello Store e testi tutti i metodi richiesti.

Esercizio 2: Gestione di una biblioteca Progettate un sistema per la gestione di una biblioteca. La biblioteca contiene diverse categorie di elementi, tra cui:

1. LibraryItem: un elemento base della biblioteca con titolo, autore, anno di pubblicazione e codice identificativo.
2. Book: un libro che eredita da LibraryItem e ha anche un editore, un ISBN e un numero di pagine.
3. Magazine: una rivista che eredita da LibraryItem e ha anche un numero di pubblicazione e una frequenza di pubblicazione.
4. DVD: un DVD che eredita da LibraryItem e ha anche un regista, una durata e un formato.
5. AudioBook: un audiolibro che eredita da LibraryItem e ha anche un narratore e una durata.

Ogni categoria di elemento deve avere i seguenti metodi:

1. LibraryItem:
   * getItemInfo(): restituisce una stringa con le informazioni dell'elemento (titolo, autore, anno di pubblicazione, codice identificativo).
   * isAvailable(): restituisce true se l'elemento è disponibile per il prestito, false altrimenti.
2. Book:
   * getBookDetails(): restituisce una stringa con i dettagli del libro (editore, ISBN, numero di pagine).
   * getBookInfo(): restituisce una stringa con le informazioni del libro (titolo, autore, anno di pubblicazione, codice identificativo, dettagli).
3. Magazine:
   * getMagazineDetails(): restituisce una stringa con i dettagli della rivista (numero di pubblicazione, frequenza di pubblicazione).
   * getMagazineInfo(): restituisce una stringa con le informazioni della rivista (titolo, autore, anno di pubblicazione, codice identificativo, dettagli).
4. DVD:
   * getDVDDetails(): restituisce una stringa con i dettagli del DVD (regista, durata, formato).
   * getDVDInfo(): restituisce una stringa con le informazioni del DVD (titolo, autore, anno di pubblicazione, codice identificativo, dettagli).
5. AudioBook:
   * getAudioBookDetails(): restituisce una stringa con i dettagli dell'audiolibro (narratore, durata).
   * getAudioBookInfo(): restituisce una stringa con le informazioni dell'audiolibro (titolo, autore, anno di pubblicazione, codice identificativo, dettagli).

Inoltre, create una classe Library che gestisca un elenco di elementi e fornisca i seguenti metodi:

* addItem(LibraryItem): aggiunge un elemento alla biblioteca.
* removeItem(LibraryItem): rimuove un elemento dalla biblioteca.
* searchByTitle(String): cerca un elemento per titolo e restituisce le informazioni dell'elemento.
* searchByAuthor(String): cerca tutti gli elementi di un determinato autore e restituisce le informazioni degli elementi.
* displayAllItems(): stampa le informazioni di tutti gli elementi della biblioteca.

Infine, create un programma principale che istanzi diversi tipi di elementi, li aggiunga all'elenco della Library e testi tutti i metodi richiesti.

Esercizio 3: Gestione di un'agenzia di viaggi Progettate un sistema per la gestione di un'agenzia di viaggi. L'agenzia offre diverse tipologie di viaggi, tra cui:

1. Trip: un viaggio base con destinazione, data di partenza, data di ritorno e prezzo.
2. Flight: un volo che eredita da Trip e ha anche un numero di volo, una compagnia aerea e una classe (economy, business, first).
3. Hotel: un soggiorno in hotel che eredita da Trip e ha anche un nome dell'hotel, una categoria (stelle) e un tipo di camera.
4. Package: un pacchetto viaggio che eredita da Trip e ha anche un elenco di attrazioni incluse e un nome del pacchetto.
5. Cruise: una crociera che eredita da Trip e ha anche un nome della nave, un itinerario e un tipo di cabina.

Ogni tipologia di viaggio deve avere i seguenti metodi:

1. Trip:
   * getTripInfo(): restituisce una stringa con le informazioni del viaggio (destinazione, date, prezzo).
   * calculateDuration(): calcola la durata del viaggio in giorni.
2. Flight:
   * getFlightDetails(): restituisce una stringa con i dettagli del volo (numero di volo, compagnia aerea, classe).
   * getFlightInfo(): restituisce una stringa con le informazioni del volo (destinazione, date, prezzo, dettagli).
3. Hotel:
   * getHotelDetails(): restituisce una stringa con i dettagli dell'hotel (nome, categoria, tipo di camera).
   * getHotelInfo(): restituisce una stringa con le informazioni del soggiorno in hotel (destinazione, date, prezzo, dettagli).
4. Package:
   * getPackageDetails(): restituisce una stringa con i dettagli del pacchetto (attrazioni incluse, nome del pacchetto).
   * getPackageInfo(): restituisce una stringa con le informazioni del pacchetto viaggio (destinazione, date, prezzo, dettagli).
5. Cruise:
   * getCruiseDetails(): restituisce una stringa con i dettagli della crociera (nome della nave, itinerario, tipo di cabina).
   * getCruiseInfo(): restituisce una stringa con le informazioni della crociera (destinazione, date, prezzo, dettagli).

Inoltre, create una classe TravelAgency che gestisca un elenco di viaggi e fornisca i seguenti metodi:

* addTrip(Trip): aggiunge un viaggio all'elenco.
* removeTrip(Trip): rimuove un viaggio dall'elenco.
* searchByDestination(String): cerca tutti i viaggi per una determinata destinazione e restituisce le informazioni dei viaggi.
* searchByDateRange(Date, Date): cerca tutti i viaggi che rientrano in un intervallo di date specifico e restituisce le informazioni dei viaggi.
* displayAllTrips(): stampa le informazioni di tutti i viaggi offerti dall'agenzia.

Infine, create un programma principale che istanzi diverse tipologie di viaggi, li aggiunga all'elenco della TravelAgency e testi tutti i metodi richiesti.

Esercizio 4: Gestione di un sistema di prenotazione di eventi Progettate un sistema per la gestione di un sistema di prenotazione di eventi. Il sistema gestisce diverse tipologie di eventi, tra cui:

1. Event: un evento base con nome, data, ora, luogo e prezzo del biglietto.
2. Concert: un concerto che eredita da Event e ha anche un artista, un genere musicale e una durata.
3. SportEvent: un evento sportivo che eredita da Event e ha anche un tipo di sport, le squadre partecipanti e una durata.
4. Conference: una conferenza che eredita da Event e ha anche un elenco di relatori, un tema e una durata.
5. Exhibition: una mostra che eredita da Event e ha anche un elenco di artisti, un tipo di mostra (arte, fotografia, ecc.) e una durata.

Ogni tipologia di evento deve avere i seguenti metodi:

1. Event:
   * getEventInfo(): restituisce una stringa con le informazioni dell'evento (nome, data, ora, luogo, prezzo del biglietto).
   * getTicketPrice(): restituisce il prezzo del biglietto dell'evento.
2. Concert:
   * getConcertDetails(): restituisce una stringa con i dettagli del concerto (artista, genere musicale, durata).
   * getConcertInfo(): restituisce una stringa con le informazioni del concerto (nome, data, ora, luogo, prezzo del biglietto, dettagli).
3. SportEvent:
   * getSportEventDetails(): restituisce una stringa con i dettagli dell'evento sportivo (tipo di sport, squadre partecipanti, durata).
   * getSportEventInfo(): restituisce una stringa con le informazioni dell'evento sportivo (nome, data, ora, luogo, prezzo del biglietto, dettagli).
4. Conference:
   * getConferenceDetails(): restituisce una stringa con i dettagli della conferenza (relatori, tema, durata).
   * getConferenceInfo(): restituisce una stringa con le informazioni della conferenza (nome, data, ora, luogo, prezzo del biglietto, dettagli).
5. Exhibition:
   * getExhibitionDetails(): restituisce una stringa con i dettagli della mostra (artisti, tipo di mostra, durata).
   * getExhibitionInfo(): restituisce una stringa con le informazioni della mostra (nome, data, ora, luogo, prezzo del biglietto, dettagli).

Inoltre, create una classe EventBookingSystem che gestisca un elenco di eventi e fornisca i seguenti metodi:

* addEvent(Event): aggiunge un evento all'elenco.
* removeEvent(Event): rimuove un evento dall'elenco.
* searchByName(String): cerca un evento per nome e restituisce le informazioni dell'evento.
* searchByDate(Date): cerca tutti gli eventi che si svolgono in una determinata data e restituisce le informazioni degli eventi.
* displayAllEvents(): stampa le informazioni di tutti gli eventi nel sistema di prenotazione.

Infine, create un programma principale che istanzi diverse tipologie di eventi, li aggiunga all'elenco dell'EventBookingSystem e testi tutti i metodi richiesti.